

# Usabilidade no ciclo de desenvolvimento de software educativo

Alex Sandro Gomes<sup>1</sup>, Stephania Padovani<sup>2</sup>

<sup>1</sup> Centro de Informática – UFPE – Caixa Postal 7851 – 50732-970 – Recife – PE – Brazil | asg@cin.ufpe.br

<sup>2</sup> Departamento de Design | UFPE | Rua Acadêmico Hélio Ramos, s/no. Cidade Universitária, Recife - PE 50740-530 | s\_padovani2@yahoo.co.uk

**Abstract.** *This course proposes a user-centred approach for the development and evaluation of educational software. First of all, we provide an overview of educational technology usage in teaching and learning environments and introduce educational software taxonomy. After that, we discuss the issue of software acceptability by its end users (students and teachers) and emphasize that an iterative learnware design cycle should include both the application of usability principles and user involvement. Finally, we propose an educational software design process and demonstrate its application through a case study.*

**Resumo.** *O curso apresenta uma abordagem centrada no usuário para o desenvolvimento e avaliação de software educativo. Inicialmente, tecemos um breve panorama da utilização de tecnologias educacionais no processo de ensino aprendizagem e apresentamos uma taxonomia de software educativo. Em seguida, discutimos a problemática da aceitação de software educativo junto a seus usuários (alunos e professores) e defendemos a aplicação de princípios de usabilidade e o envolvimento do usuário no ciclo de design iterativo de learnware. Por fim, propomos um processo de desenvolvimento de software educativo e apresentamos sua aplicação em um estudo de caso.*

## 1 Introdução

Durante muitos anos, o foco no desenvolvimento de *software educativo* parecia estar centrado no projeto e criação da interface pela interface, conforme afirmam MacDogall e Squire (1994). Em vários produtos, percebia-se que apenas aspectos relativos ao processo de aprendizagem eram observados na orientação dos projetos. Em outros produtos, ficava evidente que os projetos eram bastante orientados pela representação que os programadores tinham desse tipo de material. Podem-se associar esses fatores, dentre outros, ao fato de uma grande proporção de soluções disponíveis no mercado ser composta de itens de má qualidade.

A maior parte dos sistemas parece ser criada a partir de proposições *ad hoc* de especialistas. Analisando as publicações do WIE nos últimos anos, Gomes, Tedesco e Gomes, Tedesco e Castro-Filho (2003) mostram que mesmo as soluções educativas geradas em centros de pesquisa tendem a abordar o problema do desenvolvimento de aplicações educativas de forma não sistemática.

O importante de um software educativo é servir para que alunos e professores possam agir produtivamente sob a interface. De acordo com Hanna *et al.* (1998):

*“The best software, like the best play material, should provide a tool that allows children to explore the world creatively, using their imaginations to manipulate and assimilate knowledge about the world around them”.*

Neste sentido, as interfaces devem ser abertas à manipulação e permitir ações que promovam o desenvolvimento dos usuários. Sob uma perspectiva construtivista, as ações devem favorecer a emergência de esquemas mentais que mobilizem uma rica coleção de propriedades de conceitos do domínio.

O projeto de sistemas de softwares educativos, ou ambientes virtuais de ensino, deveria ser abordado de forma sistemática integrando aspectos de diversas dimensões de fatores humanos numa abordagem que incorpore aspectos teóricos e técnicos (Tchounikine, 2002). O processo de criação de softwares educativos demanda a identificação dos conhecimentos que emergem nas ações dos usuários com as interfaces, que ocorrem de forma não sistemática (Gomes, 1999). Para se ter uma visão sistemática acerca da relação entre o uso da interface e da aprendizagem é necessário aplicar modelos teóricos de análise que descrevam o processo de organização de ações e atividades com a interface de forma inteligível. Essa modelagem deve ocorrer em termos de elementos teóricos que revelam a adaptação progressiva dos usuários à interface e os conhecimentos que emergem durante o uso.

Neste artigo, detalhamos uma proposta de processo cíclico de desenvolvimento de software educativo centrado no usuário. Na criação de software educativo, faz-se necessário que o projeto seja orientado por aspectos do processo de aprendizagem dos conceitos visados e da prática de ensino. Deve-se observar a identificação de requisitos não funcionais para que seu uso leve os usuários a um determinado esforço cognitivo produtivo através da interação com o software, até que eles aprendam.

## **2 Software educativo**

*Inicialmente, conceituamos software educativo e apresentamos uma classificação de seus tipos. A partir de então, concentramo-nos na aceitação de software educativo junto à comunidade de usuários, abordando os benefícios do uso dessas interfaces no processo de aprendizagem e as críticas quanto à qualidade desses materiais. Apresentamos, por fim, alguns critérios que podem ser utilizados para a avaliação da qualidade de um software educativo.*

### **2.1 Conceituação e taxonomia**

Definimos software educativo como sendo um sistema computacional e interativo, intencionalmente concebido para facilitar a aprendizagem de conceitos específicos, por exemplo, conceitos matemáticos ou científicos. Na interface de uma aplicação deste tipo, conceitos são representados metaforicamente e as ações realizadas sobre os elementos da interface fazem refletir sobre os conceitos ou suas propriedades.

Segundo Lyra *et al.* (2003), utilizam-se diversos critérios na classificação de softwares educativos. Estes sistemas podem ser classificados de acordo com a liberdade de criação de situações pelos professores (aberto ou fechado), o nível de aprendizagem do aluno (seqüencial, relacional ou criativo), ou seus objetivos pedagógicos (tutoriais, aplicativos,

programação, exercícios e prática, multimídia e internet, simulação e jogos). Neste artigo priorizaremos a taxonomia relativa aos objetivos pedagógicos, proposta por Valente (1998), apresentando de forma sucinta cada uma de suas categorias.

**Tutoriais:** apresentam a informação em uma sequência pedagógica rígida, apesar de o aluno poder selecionar certas informações dentro da base de dados. A interação do aluno se restringe normalmente à leitura de textos ou a assistir vídeos e animações de interatividade reduzida.

**Exercício e prática:** apresentam lições com conteúdo e exercícios de verificação da aprendizagem. As atividades normalmente se concentram no fazer e memorizar informações e o aluno só pode mudar de estágio quando o resultado de suas atividades anteriores for avaliado pelo sistema.

**Ambientes de programação:** nestes ambientes o próprio aluno programa o computador. A realização de um programa exige que o aluno processe informações, transforme-as em conhecimento e as recodifique na sua transmissão ao sistema através da programação.

**Aplicativos:** são programas com objetivos específicos (processamento de texto, montagem e cálculo em planilhas, gerenciamento de bancos de dados, preparação de apresentações visuais), não necessariamente criados com ênfase educativa. Podem, entretanto, ser utilizados na escola de forma proveitosa para desenvolver diversas habilidades no aluno em formação.

**Multimídia e internet:** estes ambientes são utilizados principalmente com o propósito de buscar informações que serão posteriormente utilizadas em atividades educativas. O nível de interatividade dos sistemas, assim como a característica das mídias varia consideravelmente de acordo com o gênero e a estratégia do sistema.

**Simulações:** simulam a ocorrência de fenômenos no computador. Há sistemas mais fechados em que o usuário simplesmente assiste à simulação ou escolhe que simulações deseja assistir e outros mais interativos e ricos onde o próprio aluno pode criar o modelo do fenômeno, alterar parâmetros e então assistir à simulação resultante.

**Jogos:** são sistemas de entretenimento em que o usuário tem suas habilidades cognitivas constantemente desafiadas, motivando o aluno a resolver problemas que envolvam a aplicação de conhecimentos prévios. A competição pode ocorrer entre o aluno e o sistema ou entre os próprios alunos, sendo mediada pelo computador.

## **2.2 Aceitação de software educativo junto aos usuários**

Ainda no início dos anos 90, Carraher (1990) concluía de forma pessimista quanto às expectativas de surgimento, num tempo breve, de softwares educativos de boa qualidade pedagógica. O autor observava à época que:

*“a produção de software de qualidade técnica e, mais ainda, de qualidade pedagógica, é mais complexa do que se imaginava há pouco tempo atrás, de tal modo que dificilmente surgirá, de forma rápida e espontânea uma quantidade de software de qualidade.”* (Ibid., p. 36).

A literatura internacional sobre a qualidade de software educativo tem questionado a qualidade desses softwares que parecem não atender às expectativas dos profissionais de

ensino (Hinostroza e Mellar, 2001). O impacto epistemológico do uso dessas novas tecnologias nos processos de ensino e aprendizagem é menor que o esperado pelos professores (Balacheff e Kaput, 1996; Schwartz, 1999).

No contexto educacional há uma defasagem entre as expectativas geradas a partir do potencial das tecnologias digitais e interativas e o modo como estão inseridas e mediando as atividades pedagógicas (Resnick, 2001; Dugdale, 1999). Muitas dessas defasagens devem-se às deficiências de usabilidade, geradas da inadequação dos métodos e operações que muitos desses instrumentos materializam com a cultura docente sedimentada no cotidiano da profissão.

Uma outra frente de problemas é aberta quando consideramos a qualidade dos softwares educativos (Frye e Soloway, 1987; Hinostroza e Mellar, 2001;). De acordo com Hinostroza e Mellar (2001), os designers de softwares educativos priorizaram as questões relacionadas com a aprendizagem em detrimento das atividades de ensino, e isto seria uma das causas das dificuldades que os professores sentem ao tentarem incorporar estes materiais às suas atividades em sala de aula.

### **2.3 Qualidade de software educativo**

A qualidade de um software educativo está relacionada com a capacidade que o mesmo tem de atender a requisitos e necessidades do usuário relacionadas à aprendizagem. Para tal, faz-se necessário definir métricas para cada variável pertinente. A qualidade de um software educacional depende do processo de seu desenvolvimento voltado à produção de um artefato que atenda aos objetivos e necessidades dos usuários (Rocha e Campos, 1993).

Os usuários mais representativos podem ser alunos, professores ou mesmo a instituição escolar. De uma forma geral, a qualidade de um software educativo está relacionado com a finalidade de uso e com a possibilidade de a mesma permitir a emergência de situações que levem os usuários a um nível de esforço mental que os façam compreender parte das propriedades de um novo conceito. O entendimento das necessidades do usuário é visto como um fator de sucesso para o desenvolvimento de um produto.

Este entendimento deve ocorrer no início do desenvolvimento, quando os requisitos são definidos, caso contrário o sistema não será bem aceito pelos usuários, gerando a necessidade de um re-trabalho para que venha atender às reais necessidades. Segundo Maguire (2001), um ciclo de qualidade de software exige o envolvimento do usuário. A qualidade do processo de desenvolvimento de software pode ser melhorada através do uso da ISO 13407 e ISSO TR 18529, que definem atividades centradas no usuário.

Lins e Gomes (2003) mostraram que há um conflito entre a atividade proposta pelo professor e as atividades propostas pelos softwares. No entanto, abordar a qualidade sob o ponto de vista da prática docente é tão limitado quanto focar em estudos da característica da interface. McDougall e Squires (1994) afirmam que o projeto de um software educativo deveria contemplar, com igual relevância, aspectos relativos a três domínios: currículo embutido no design, aspectos relativos à aprendizagem do aluno no domínio visado (epistemologia do domínio) e aspectos do contexto de uso (sendo mais representativos os professores e alunos usuários). A maioria das aplicações foca excessivamente em aspectos relativos a aprendizagem dos alunos.

Um outro critério de qualidade atribui ao ciclo de desenvolvimento de software a origem da qualidade de um material. A noção de qualidade é fundamental para definir níveis de facilidade ou acessibilidade dos usuários às funções do sistema. Segundo essa perspectiva, a noção de usabilidade é fundamental para a definição de um sistema de qualidade. Os benefícios da usabilidade incluem aumento de produtividade, melhoria da qualidade de trabalho, redução nos custos de suporte e treinamento e aumento da satisfação do usuário. A usabilidade é atingida a partir da forma como ocorre o processo de construção da aplicação, desde o projeto do sistema.

### **3 Usabilidade no projeto de software educativo**

*Nesta etapa, apresentamos definições para o termo usabilidade, assim como a operacionalização deste conceito em critérios para avaliação de uma interface computadorizada. Listamos e descrevemos, então, alguns conjuntos de princípios de usabilidade propostos por autores das áreas de ergonomia, interação humano-computador e design. Por fim, discutimos as diversas abordagens de avaliação de usabilidade de softwares educativos.*

#### **3.1 Conceituação e critérios de avaliação de usabilidade**

O termo usabilidade é normalmente encontrado na literatura se referindo a um conjunto de propriedades que deve ser avaliado em determinada interface, ou para definir uma qualidade positiva que essa interface deve possuir. Entretanto, é importante ressaltar que quando compradores e usuários tomam a escolha de um sistema, sua decisão não depende apenas da usabilidade desse sistema, mas sim de uma avaliação mais abrangente que envolve a consideração de diversos fatores:

- utilidade (o sistema atingirá os objetivos necessários em termos funcionais?);
- compatibilidade (o sistema será compatível com outros sistemas já em uso?);
- aceitabilidade (os usuários perceberão que o sistema é adequado?);
- custos econômicos (quais são os custos de aquisição e de manutenção?);
- custos sociais (quais são as consequências sociais e para a organização?).

Portanto, a usabilidade é uma preocupação específica dentro de uma preocupação maior que é a aceitabilidade prática e social do sistema. Mas o que exatamente entendemos por usabilidade? Há várias tentativas de se encontrar uma definição precisa para o termo, cada uma enfocando diferentes aspectos da interação usuário-sistema. Alguns exemplos são:

*“Usability is presented as a concept which can limit the degree to which a user can realize the potential utility of a computer system.” (Eason, 1984, p.133)*

*“ Usability concerns the extent to which an end-user is able to carry out required tasks successfully, and without difficulty, using the computer application system. ” (Ravden & Johnson, 1989, p. 09)*

*“ Usability is the capability in human functional terms to be used easily and effectively by the specified range of users, given specified training and*

*user support, to fulfil the specified range of tasks, within the specified range of environmental scenarios. ” (Shackel, 1991, p. 24)*

*“ The usability of a computer is measured by how easily and how effectively the computer can be used by a specific set of users, given particular kinds of support, to carry out a fixed set of tasks, in a defined set of environments. ” (Chapanis, 1991, p. 362-363)*

*“ Usability is a quality attribute that assesses how easy user interfaces are to use. The word ‘usability’ also refers to methods for improving ease-of-use during the design process. ” (Nielsen, 2003)*

Entretanto, ao explorarmos as definições de usabilidade apresentadas, verificamos que nenhuma provê indicação do processo necessário para se conduzir uma avaliação de usabilidade. Vários autores reconhecem que os termos ‘facilmente’ e ‘eficazmente’ são bastante imprecisos e buscam critérios mais operacionais com base em erros, tempo de execução, tempo de aprendizado, avaliações subjetivas dos usuários, entre outros.

Apresentamos a seguir uma combinação dos critérios operacionais propostos por Chapanis (1991), Shackel (1991) e Nielsen (1995), ou seja as formas propostas pelos autores para se definir facilidade de uso e eficácia de maneira observável e mensurável.

- facilidade de inicialização (tempo necessário para se instalar o programa e iniciar sua utilização);
- facilidade de aprendizado (tempo necessário para aprender um conjunto básico de operações que permitem ao usuário iniciar suas tarefas);
- facilidade de memorização (o sistema deve ser fácil de lembrar, de maneira que o usuário ocasional seja capaz de reutilizá-lo sem ter que reaprendê-lo).
- eficácia (o conjunto de tarefas especificado deve ser completado acima do patamar de performance definido);
- eficiência (uma vez que o usuário aprendeu a utilizar o sistema, um alto nível de produtividade é atingido);
- taxa de erros (quantidade de erros cometidos e tempo para corrigi-los);
- versatilidade (número de diferentes funcionalidades que o sistema oferece);
- flexibilidade (o sistema deve permitir variações na forma de realização da tarefa e no ambiente de realização);
- atitude (a tarefa deve ser realizada dentro de níveis definidos como aceitáveis em termos de cansaço, desconforto, frustração e esforço pessoal);
- satisfação (uso continuado do sistema de forma estimulada).

Mais recentemente, alguns autores da área de interação-humano computador passaram a questionar o foco excessivo da usabilidade na melhoria da eficiência e da produtividade no trabalho. Preece et al. (2002) argumentam que o design de sistemas computadorizados deveria se preocupar também com a experiência do usuário, ou seja, em como o usuário se sente ao interagir com o sistema.

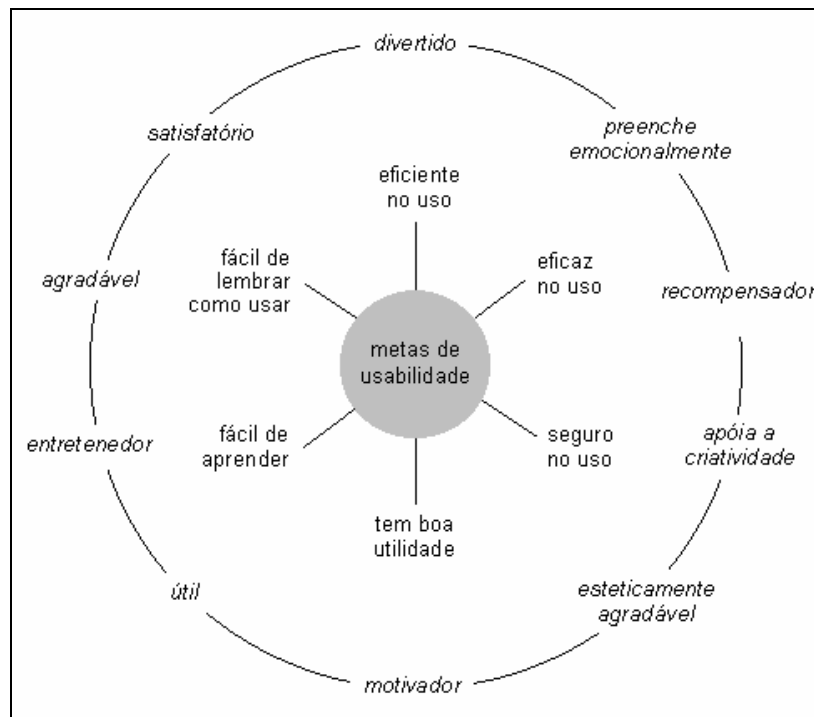


Figura 1: metas de usabilidade [dentro do círculo] e metas de experiência [sobre a circunferência] | Fonte: traduzido e adaptado de Preece et al., 2002: 19.

Os autores que seguem essa tendência se preocupam principalmente em criar sistema que além de fáceis de usar sejam divertidos, prazerosos e esteticamente agradáveis. As pesquisas tentam investigar a questão do prazer obtido na interação de forma mais detalhada. Os resultados apontam alguns aspectos que contribuem para uma interação prazerosa: atenção, ritmo, jogabilidade, interatividade, controle consciente e inconsciente, engajamento e estilo de narrativa empregados no projeto do sistema.

Convém ressaltar que a avaliação de usabilidade em software educativo deve ser guiada não somente por critérios ergonômicos como os que são discutidos nos estudos de Gamez (1998), mas também por aspectos pedagógicos (programas de ensino, objetivos, formas de avaliação), cognitivos (forma de aquisição do conhecimento, a maneira como o conhecimento é guardado na memória), psico-pedagógicos (motivação, individualização da aprendizagem), lúdicos (referentes ao caráter de jogos, brinquedos e divertimentos) e sócio-culturais (oportunidade de uso do computador, intercâmbio cultural, questões associadas à cultura do usuário e sua formação), conforme argumenta Gladcheff (2001).

### 3.2 Princípios de usabilidade

Uma outra forma de se considerar a usabilidade é em termos de princípios de design. Esses princípios são abstrações genéricas cujo objetivo é orientar designers durante o desenvolvimento de sistemas. Estes princípios são derivados de conhecimento teórico, resultados de pesquisas empíricas e, por que não dizer, senso comum. Eles tendem a ser redigidos de forma prescritiva, sugerindo aos designers o que fazer e o que evitar em

suas interfaces, mas sem a intenção de trazer recomendações sobre a forma específica que os elementos da interface devem tomar.

Apresentamos a seguir uma comparação dos princípios de usabilidade propostos por Nielsen (1995), Mayhew (1992), Shneiderman (1998), Marcus (1997) e Bastien e Scapin (1993) para nortear o projeto de sistemas computadorizados interativos. Apesar de utilizarem expressões diferenciadas, verifica-se o consenso entre esses autores no que diz respeito aos princípios mais relevantes, como simplicidade, compatibilidade com o usuário, consistência, feedback, minimização da carga na memória de curta duração, prevenção e fácil correção de erros, controle e flexibilidade. Muitos desses princípios serviram, mais recentemente, de base para a construção de *checklists* de desenvolvimento e avaliação de softwares educativos.

	autores que promovem o princípio				
princípios	Nielsen	Mayhew	Shneiderman	Marcus	Bastien & Scapin
simplicidade	X	X		X	X
clareza no diálogo	X		X	X	
compatibilidade com o usuário	X	X	X	X	X
compatibilidade entre produtos		X			
compatibilidade com a tarefa		X		X	X
compatibilidade com o contexto					X
consistência	X	X	X	X	X
feedback	X	X	X	X	X
minimização da carga da memória de curta duração	X		X	X	X
mensagens de erro claras	X			X	
prevenção, proteção e fácil correção de erros	X	X	X	X	X
providenciar help online e documentação sobre o sistema	X			X	
familiaridade		X			
	autores que promovem o princípio				
princípios	Nielsen	Mayhew	Shneiderman	Marcus	Bastien & Scapin
controle do usuário sobre o sistema		X	X	X	X
flexibilidade   personalização		X	X	X	X
tecnologia invisível		X			
robustez técnica					
grupamento funcional explícito				X	X
legibilidade e leitura das informações apresentadas					X
informação contextual para a tomada de decisões				X	



códigos significativos					X
------------------------	--	--	--	--	---

Tabela 1: comparação dos princípios de usabilidade propostos por diversos autores

Fonte: Gomes & Padovani, 2005.

### 3.3 Envolvimento do usuário no projeto de software educativo

A característica principal de um projeto que leva a um produto de boa qualidade é o envolvimento do usuário, ou 'design centrado no usuário'. O objetivo de técnicas de projeto centrado no usuário é o desenvolvimento de produtos úteis e usáveis. Usabilidade é o conceito chave em como auxiliar os usuários nas suas atividades. Os benefícios da usabilidade incluem aumento de produtividade, melhoria da qualidade de trabalho, redução nos custos de suporte e treinamento e aumento da satisfação do usuário. A usabilidade será conseguida através da participação de usuários potenciais no projeto do sistema. A ISO 13407 (1999) "*Human-centered design processes for interactive systems*" sugere que a participação do usuário é essencial para que requisitos relevantes dos usuários e das tarefas possam ser identificados quando da especificação do sistema. A participação do usuário é mais eficiente se acontecer nos estágios iniciais do desenvolvimento do sistema, pois o custo envolvido em mudar durante e, principalmente, após o desenvolvimento do sistema é bem mais alto.

Concordamos com Moraes (2003) quando a autora afirma que:

*"tradicionalmente, a preocupação com a usabilidade só ocorre no final do ciclo de design, durante a avaliação do produto já finalizado. Resulta que poucas modificações são implementadas e se algumas realmente substantivas o são implicam em custos elevados. Tem-se, então, que desde o início da atividade projetual a consideração com a usabilidade – a ergonomia - deve estar presente. O modo mais óbvio de obter informações sobre as atividades da tarefa relacionadas com as comunicações e interações das pessoas com os sistemas é observá-las ao realizar tomadas de informação, acionamentos, deslocamentos, comunicações e perguntar como e porque o fazem".*

Criar um software educativo com qualidade significa observar detalhes em cada uma das etapas de produção do mesmo. Trata-se da criação de um sistema dinâmico com especificações e requisitos específicos quanto à interação com o usuário e seu impacto sobre a aprendizagem (Gomes e Wanderley, 2003). Frequentemente, interfaces são criadas a partir da especificação das tarefas que deveriam ocorrer mediante seu uso. No caso do design de um *software educativo*, a análise de tarefas é insuficiente. São necessárias a análise e a consideração do processo de ensino e aprendizagem experimentado por usuários, professores e alunos. Portanto, procedimentos de concepção deveriam ser permeados de testes de usabilidade que considerassem verificar se tais aspectos, como a aprendizagem de conceitos específicos, o currículo e a estrutura da prática de ensino, são implementados na interface.

Do ponto de vista dos usuários finais de um software educativo, os alunos, a principal meta do sistema seria o aprendizado semântico. Entretanto, para que esse aprendizado semântico ocorra, há a necessidade de aprendizado sintático, ou seja, aprender como utilizar as ferramentas do sistema para realizar então as atividades educativas que

levarão ao aprendizado semântico. Neste contexto, torna-se evidente a importância da facilidade de diálogo entre o usuário e o sistema computadorizado, pois caso a operação do sistema seja algo dificultoso para o usuário, esta atividade competirá por recursos cognitivos com a tarefa principal de aprendizado de conteúdos, degradando a performance em pelo menos uma das tarefas. Da mesma forma, um diálogo dificultoso pode diminuir o interesse do usuário pelo software, mesmo que seu conteúdo lhe interesse, prejudicando o aprendizado semântico.

Porém, antes que o software educativo esteja sendo utilizado pelos alunos, houve todo um trabalho de planejamento, realizado por seus usuários indiretos: os professores. Para esses usuários, é necessário que o sistema seja fácil de instalar, de operar e que forneça instruções claras sobre como integrá-lo nas atividades didáticas de ensino-aprendizagem. Carraher (1990) argumenta que um software educativo por si só não é um “estimulante” para o aprendizado, pois seu uso precisa estar apoiado numa integração entre os envolvidos e as demais tarefas do contexto.

Mas nem sempre o software estará sendo utilizado em sala de aula. Ocorrerão situações em que o mesmo será levado para a casa do aluno e parentes também atuarão como facilitadores ou co-usuários. Para estes usuários ocasionais, é importante que o sistema seja de fácil instalação e que as ferramentas do software sejam de fácil entendimento e memorização, pois assim o parente poderá auxiliar o aluno nas atividades escolares sem ter que lidar com dificuldades operacionais.

Verifica-se, portanto, a necessidade da participação de todos os *stakeholders* envolvidos na utilização do software educativo. Um processo de design iterativo que inclua e integre a participação de professores, orientadores, pedagogos e alunos, certamente trará como resultado um sistema mais compatível com o perfil desses usuários diretos e indiretos e, conseqüentemente, uma melhor usabilidade.

### **3.4 Métodos de avaliação de usabilidade aplicáveis a software educativo**

A avaliação é uma atividade muito importante no ciclo de design iterativo. Nos últimos anos, a comunidade de ihc tem desenvolvido e testado uma ampla gama de métodos práticos, alguns dos quais podem ser aplicados a softwares educativos. Existem atualmente, inclusive, ferramentas de avaliação dirigidas a alunos, pais e professores disponíveis online. Comunidades educativas também criaram o hábito de disponibilizar em seus sites os resultados de avaliações de software educativo para dar suporte a professores na escolha de material educativo (ex.: UNESCO ICT in Education, NETS for Teachers, Southern Region Educational Board – USA, SuperKids Educational Software Review).

Na área de usabilidade há diferentes abordagens para a avaliação de interfaces computadorizadas. Cada abordagem adota seus próprios critérios, procedimentos, foco e tem, conseqüentemente, maior ou menor aplicabilidade dependendo do sistema em processo de avaliação. Também há diferenças no que concerne à expertise técnica, custos, ambiente e equipamentos necessários à condução da avaliação.

Lansdale & Ormerod (1995) propõem quatro categorias para caracterizar as abordagens de avaliação de usabilidade: testes de usabilidade; guidelines, standards e checklists; modelagem formal; e inspeção por especialistas. Essas mesmas categorias aparecem nas classificações de Chapanis (1991), Shackel (1991), Christie et al. (1995), Dix et al.

(1998) apesar de utilizarem nomenclaturas um pouco diferentes e alguns dos autores sugerirem também sub-categoriais.

**Testes de usabilidade:** Esta abordagem se baseia no princípio de produção de protótipos e captura de dados observando a interação dos usuários com o sistema. Para que os resultados desses testes sejam válidos, é importante recrutar usuários representativos do público alvo e escolher tarefas realistas para serem realizadas durante os testes. Também é importante decidir que tipo de dados se deseja obter e como a análise será realizada como parte do planejamento dos testes.

**Guidelines, standards e checklists:** Guidelines são recomendações sistematicamente organizadas para orientar o design ou a avaliação de certos aspectos da interface. Standards são guidelines exaustivamente discutidos por instituições especializadas que se tornam então normas. Checklists são baseados em guidelines e standards e consistem em uma série de questões agrupadas em categorias, cujas respostas dão um indicativo da qualidade do software. Essa abordagem é composta de ferramentas de transferência, pois especialistas em ergonomia e usabilidade formulam guidelines e checklists que podem ser aplicados por desenvolvedores e avaliadores não especialistas.

**Modelagem formal:** O objetivo desta abordagem é definir propriedades de interação da interface e identificar as habilidades cognitivas necessárias para sua utilização, prevendo a performance do usuário. A modelagem pode ser utilizada para indicar pontos críticos da interface onde a ocorrência de erros e incompatibilidades é mais provável e então optar por soluções particulares de design. Alguns exemplos de métodos de modelagem são GOMS, KLM, TAG e HTA.

**Inspecção por especialistas:** Consiste na emissão de pareceres sobre a qualidade do software com base em um conjunto de princípios (heurísticas) ou pela simulação da realização de uma série de tarefas (abordagem *walkthrough*). Esses especialistas podem ser da área de usabilidade, cujo propósito maior é inspecionar a interface em termos de sua facilidade de uso, ou especialistas no domínio, os quais avaliam a correção do conteúdo, estruturação e atividades disponibilizadas aos usuários. Os métodos mais comuns são a avaliação heurística e a abordagem *walkthrough*.

No contexto da avaliação de usabilidade de interfaces educativas, somam-se às categorias anteriores três perspectivas que orientam a realização de testes de usabilidade. Cada uma delas distingue-se das demais pelo foco de estudo. A primeira classe corresponde aos testes de usabilidade centrados no artefato. A segunda classe focaliza sua análise na aprendizagem, mediada pela análise das adaptações cognitivas conforme modelo construtivista. A terceira classe de testes analisa os contextos sociais de uso das interfaces.

**Testes centrados no artefato:** A primeira classe de testes de usabilidade engloba os testes que permitem descrever as interfaces sem que haja necessidade de interpretar fatores implicados em processos cognitivos. Nessa classe encontram-se os testes do tipo checklist, ou usabilidade *guideline*, e testes baseados em princípios de design (Norman, 1988).

**Testes centrados na aprendizagem:** A segunda classe de testes de usabilidade relaciona testes que descrevem o uso da interface pelo usuário, com o foco da análise na ação do usuário. O modelo teórico é de base construtivista e a unidade central de análise

é o conceito piagetiano de esquema mental. Em alguns testes a ação relaciona-se com aprendizagem (Gomes, 1999). Esse modelo permite analisar a aprendizagem de conceitos que ocorre ao longo do processo de adaptação do usuário ao uso da interface. Diferente da abordagem anterior, podem-se analisar estratégias de realização de tarefas e a aprendizagem mediada em processos de adaptação à análise da aprendizagem mediana.

**Testes centrados no contexto social:** A terceira classe de testes analisa a afinidade de uso da interface, ou seja, a interação entre o usuário e a interface. A unidade de análise é a atividade, unidade mista composta de processos cognitivos e artefatos mediadores.

#### 4 Design iterativo de software educativo

*O design iterativo é uma forma simples de integrar processos de inovação – de escuta das necessidades dos usuários – no design de aplicações educativas. Existe uma grande lacuna na literatura sobre aplicações educativas, que tendem a mostrar os pontos negativos e formas de avaliação. A criação de aplicações educativas é um processo misto de pesquisa e de concepção que precisa ser entendido em sua dinâmica de práticas. Nesse curso comunicaremos o processo de design iterativo dentro da cultura de engenheiros de software educativos.*

O design iterativo apresenta-se como um processo cíclico desenvolvimento de software educativo centrado no usuário. O Design Iterativo (Nielsen, 1995) fundamenta-se na idéia de se desenvolver o sistema aperfeiçoando por aproximações sucessivas, estas submetidas a testes e refinamentos a fim de se obter um produto final que responda aos requisitos levantados na fase de contextualização do problema. Cada iteração objetiva identificar e corrigir certos problemas de usabilidade mais evidentes, que podem até vir a encobrir outros, e desse modo promover reparos no learnware.

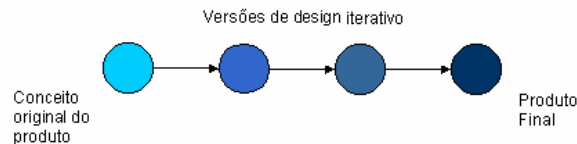


Figura 1: Estrutura de Design Iterativo.

O Design Iterativo funciona como uma alternativa a minimizar os problemas de usabilidade em software educativos, fazendo com que estes atendam as necessidades de professores e alunos no que diz respeito à aprendizagem dos conceitos matemáticos relacionados a frações.

Na criação de software educativo, faz-se necessário que o projeto seja orientado por aspetos do processo de aprendizagem dos conceitos visados e da prática de ensino. O Design Iterativo projeto de um learnware, seja de software educativo, seja de ambientes virtuais de ensino, deve ser abordado de forma sistemática em suas mais diversas dimensões. Deve-se observar a identificação de requisitos não funcionais para que seu uso leve os usuários a um determinado esforço cognitivo produtivo através da interação com o software, até que eles aprendam.

#### 4.1 Entendendo o usuário e suas atividades

Parte dos requisitos de um software educativo visa facilitar o uso dos usuários finais. Tais aplicações podem ser usadas em diversos contextos e por diferentes conjuntos de usuários. Pode ser usado por um usuário apenas, ou por mais de um usuário. Estes podem encontrar-se lado a lado ou estarem distantes uns dos outros interagindo através de ambientes virtuais (Bricker, 1993).

Alguns aspectos da interface de software educativos que interferem na atividade de sala de aula, ou em laboratório de informática. Observamos que a maioria dos softwares é construída dentro de um paradigma específico que projeta o software num contexto da atividade com apenas um usuário. Parece não haver uma preocupação em observar a atividade normal de um professor num laboratório de informática educativa para a partir dessa observação gerar requisitos que promovam o design de ferramentas adequadas, num processo conhecido como design centrado no usuário (Bevan, 1997), e desse modo, atender as necessidades de professores e alunos no contexto.

Entender a prática no contexto de uso da aplicação orienta igualmente o projeto da interface (Nardi, 1996). A partir da análise de estudo do contexto, torna-se explícita a classe de requisitos que são destinados a informar o designer de como as funções atendem as necessidades dos usuários no contexto de uso. Adota-se para tal os preceitos da Teoria da Atividade (Leont'ev, 1975) dada seu alto grau de abstração, utilizada para a modelagem da atividade de usuários e grupos de usuários. Para essa teoria qualquer atividade humana é mediada e, portanto suas estruturas são passíveis de análise. A utilização desse modelo permite expressar as relações que existem entre os elementos que estruturam uma atividade. A partir da identificação desses elementos, fica mais simples inferir requisitos que uma interface, a ser inseri no contexto da mesma atividade, deve possuir (Martins e Daltrini, 1999).

Métodos de pesquisa etnográfica ficaram populares no campo de estudo de interação usuário máquina. Estes métodos de pesquisa são usados no design de uma grande variedade de produtos (Kujala, Kauppinen e Rekola, 2001). O uso destes métodos para o desenvolvimento de produto caracteriza-se como forma de reunir exigências de clientes para avaliação de campo de uso de novos produtos e serviços. A execução destes métodos inclui trabalho de campo em condições reais, com uma perspectiva objetiva e com ricas descrições das pessoas, ambientes e interações existentes. Engloba o entendimento amplo das variáveis ambientais e do contexto de uso dos produtos que estão sendo projetados.

O estudo de Lins e Gomes (2003) foi realizado através da observação das aulas desenvolvidas pelos docentes em laboratórios de informática, ou seja, contexto natural de uso do software. Os resultados obtidos com o estudo dos episódios detectaram vários tipos de demandas por interação do professor em aulas que usam o *software* educacional: demandas por interações de alunos que não tinham familiaridade com o computador, demandas de alunos que não conseguiam navegar, no *software*, por falta de clareza das instruções do *software*; demandas por dificuldades dos alunos em relação ao conteúdo; demandas por limitações técnicas dos *softwares*; e demandas por que o aluno não entende o *feedback* do *software*. Em geral, a quantidade de demandas interferia no desenvolvimento da aula, dificultando a interação do professor. A partir dessa análise da estrutura das atividades que ocorrem em sala de aula é possível gerar requisitos para as

aplicações de forma a adequá-las. Após a análise da estrutura das atividades, procedemos novamente com a análise de necessidades associadas a cada uma das dimensões da atividade. Identificamos assim, necessidades relacionadas: aos usuários (*user needs*), as regras (*rules needs*), a comunidade (*community needs*), a divisão do trabalho (*work needs*), ao objeto (*object needs*).

#### **4.2 Requisitos do sistema a partir de uma análise do domínio**

O primeiro momento de um projeto de um software educativo corresponde à identificação do domínio a ser veiculado, em termos de conceitos ou campos conceituais. Entendemos por domínio o campo conceitual a ser apresentado na interface. O domínio de aplicação de um software pode ser distinto como geometria e linguagem de programação. Para o projeto de um software educativo são identificados os domínios desde o início do projeto. Os requisitos estão relacionados com o que a interface deve prover em termos de aprendizagem.

A identificação das necessidades relativas ao domínio tem a função de orientar na elaboração de casos de uso, funcionalidades, que estão em conformidade com a proposição de situações Adequadas à aprendizagem dos conceitos do domínio. Casos de uso são gerados de forma orientada a necessidades dos usuários elevando o número de dados sobre a aprendizagem dos usuários que é considerada na especificação do sistema. A análise da aprendizagem do domínio permite identificar requisitos associados ao domínio visado da aprendizagem quando esses não são encontrados na literatura especializada. Esta complementa ou compensa a ausência de fontes bibliográficas de dados.

As fontes de informações sobre a elicitação de requisitos relacionados ao domínio são os periódicos sobre a aprendizagem de conceitos específicos. De uma forma geral, a eficácia do software em provocar a aprendizagem “dependerá do grau em que lida com o conhecimento e com concepções que os alunos têm” (Carraher, 1990, p. 36). Esse tipo de fonte de informação já era recomendado por Carraher (*Ibid.*) que sugeria que equipes de produção de software educativo precisariam incluir especialistas em aprendizagem e desenvolvimento cognitivo, mas que tenham entendimento e domínio sobre os resultados oferecidos pelas publicações sobre aprendizagem de conceitos específicos.

Em áreas como o Ensino da Matemática e Ciências, os dados de pesquisas acerca da aprendizagem de conceitos são abundantes. Algumas áreas são menos privilegiadas em termos de resultados de pesquisa, como é o caso de áreas técnicas de formação profissionalizante. No caso da ausência de literatura suficiente acerca do processo de aprendizagem do conceito do domínio, a alternativa é a realização de estudos sistemáticos. A partir dos resultados obtidos na pesquisa bibliográfica e/ou com estudos empíricos relacionados ao domínio, pode-se sintetizar as necessidades dos usuários a serem atendidas pela interface relacionadas ao domínio.

#### **4.3 Prototipagem rápida de interfaces educativas**

Durante a realização de estudos para elicitação de requisitos e identificação de necessidades dos usuários, duas alternativas de procedimentos podem ser ponderadas à decisão. A primeiro é a realização de um estudo para análise das atividades no contexto,

sem o uso de protótipos. Uma segunda opção seria realizar estudos nas quais tenta-se construir protótipos diversos do produto final usando como artefato um sistema similar. Conforme a flexibilidade dos protótipos eles podem ser considerados leves (cenários argumentativos) ou tangíveis (feitos de papel ou de algo similar). A construção dos protótipos pode seguir diferentes níveis de fidelidade com o produto final. Uma forma de aproximar versões de baixa fidelidade do formato de produtos finais é seguindo guidelines ou guias de recomendação de plataformas ou de tipos de produtos. Existe ainda a possibilidade de criarmos personagens fictícios representativos de um usuário com características médias para imaginar como ele usaria nossos produtos. Utilizamos todas essas técnicas em nossos projetos, as quais descrevemos a seguir.

#### **4.3.1 Cenários**

O termo “Cenário” possui muitos e variados significados na literatura de diferentes áreas, desde do planejamento estratégico, engenharia de software até HCI. Esses significados têm um determinado núcleo que nos remete a uma forma de representação, em forma de narrativas, de práticas humanas. No projeto de sistemas interativos, os cenários são usados para representar formas de uso futuras e hipotéticas, antecipando a visualização de problemas que possam vir a acontecer. Cenários são caracterizados como sendo hipotéticos, seletivos, no entanto, conectados com uma realidade. Os cenários necessitam estar em harmonia com a situação que se deseja considerar.

Em um artigo recente, Bødker (2000) propõe três razões principais para elaborar cenários no design interativos de dispositivos: a possibilidade de antecipar soluções futuras situadas no presente, sua capacidade de ilustrar soluções alternativas e ainda permitir a análise dessas soluções alternativas para identificar problemas potenciais antes do produto estar terminado. O resultados das análises desses estudos geram casos de uso, cujos cenários, seqüências básicas e exceções, são descritas de tal maneira a atenderem as necessidades dos usuários em seu contexto. Pode-se também utilizar os cenários a partir de resultados de trabalhos etnográficos.

A análise de cenários de uso permite a realização de estudos iniciais da adequação de interfaces educativas (Gomes *et al.* 2002) para verificar se a mesma promove a emergência de situações favoráveis à aprendizagem de conceitos do domínio. Trata-se de uma maneira de antecipar e permitir o estudo de fatores humanos da interface para identificação de requisitos do domínio e relativos ao processo de mediação a ser promovido com a mesma.

Em um trabalho recente, Falcão (2004) utilizou a técnica de cenários seguida de uma observação etnográfica da prática de alfabetização na educação infantil para gerar protótipos tangíveis de livros interativos. A etnografia revelou as principais dificuldade e necessidade de uma professora dentro da sala de aula com muitas crianças. Cenários hipotéticos otimistas e pessimistas foram criados para antecipar o uso com um dispositivo que permitia projetar imagens de um livro. Uma grande quantidade de requisitos foi identificada durante esse processo.

#### **4.3.2 Geração de protótipos de baixa fidelidade**

Uma das formas mais eficientes de balizar o projeto de uma interface educativa é através da verificação constante no nível de aderência dos produtos aos requisitos dos

mesmos. Após a realização de cenários que representam a utilização dos sistemas – aqui considerados como protótipos leves -, um segundo tipo de protótipo pode ser realizado. Com uma maior quantidade de requisitos conhecidos realiza-se a construção e testes de versões tangíveis das interfaces. A construção produz protótipos na forma de *mockups*. Os protótipos podem ser avaliados através de métodos de análise da tarefa e da análise instrumental como descreveremos a seguir. Com base nos resultados dos testes de usabilidade, realizam-se as alterações necessárias na interface e esta é submetida em sua versão final a testes de usabilidade envolvendo especialistas e usuários finais.

Uma referência recente e bem acolhida na área de design interativo, Snyder (2003) baseia o processo de prototipagem rápida em sua experiência à frente de dezenas de projetos executados no contexto de mercado. A autora descreve um processo para realização de ciclos de prototipagem rápida definindo o que acontece nos diversos momentos dessa atividade, quem são os responsáveis e o tempo previsto em cada atividade. De uma forma geral, os ciclos de prototipagem rápida são compostos de:

**Reunião inicial** – Nesse encontro se discutem objetivos, riscos e preocupações. Todos os participantes do projeto devem participar deste encontro. Por conta da diversidade de profissionais envolvidos em um projeto como esse, diversas reuniões são necessárias para permitir um alinhamento de vocabulário e a definição de instrumentos que vão permitir a comunicação e a fluidez do trabalho coletivo.

**Recrutamento de usuários** – Essa atividade segue as recomendações básicas de recrutamento de usuários de testes proposta em Nielsen (1995). É importante destacar também que esse recrutamento é parte do controle da metodologia de pesquisa que se utiliza nesses testes.

**Design de tarefas** – Para a realização dos testes de usabilidade especificam-se tarefas que serão executadas. Deve participar desta atividade o núcleo do time de desenvolvimento, auxiliado por alguém que tenha um bom conhecimento sobre o domínio daquilo que deve ser testado. A definição das tarefas deve aproximar-se do que o usuário encontrará como tarefa quando o sistema estiver construído. Isso antecipa a captura de feedback junto aos usuários em etapas ainda iniciais do projeto.

**Criação do protótipo e *walkthroughs*** – A partir da definição dos elementos das interfaces que devem ser testadas são elaboradas as interfaces. Utilizando uma técnica de grupo conhecida como *cognitive walkthroughs* – que simula a utilização do material pelos usuários – é possível para equipe refinar os modelos e criar sequência de ações (script) sem a participação de usuários reais, mesmo antes dos testes de usabilidade.

**Testes de usabilidade e refinamento iterativo** – Com os materiais e os scripts definidos são executados os testes de usabilidade. A coleta pode ocorrer por meio de gravações de áudio e/ou vídeo e posteriormente os dados são analisados. É de boa prática usar de anotações durante a coleta de dados com os usuários. A análise dos dados permite listar os problemas encontrados após cada teste e revisar o protótipo antes do próximo teste.

**Priorização de problemas e plano de ação** - Nesta atividade, os problemas que não foram resolvidos são priorizados, aqueles mais graves são discutidos (bem como possíveis soluções), um plano de ação para resolver os problemas encontrados é elaborado e os mesmos são listados.



**Comunicação de resultados** – É nessa etapa que a equipe de desenvolvimento tem a possibilidade de aprender e apreender a forma de agir do usuário. Um ou dois membros da equipe (geralmente, mas não necessariamente, os especialistas em usabilidade) devem escrever um sumário dos dez maiores problemas descobertos, disponibilizar os resultados para a equipe de desenvolvimento, redigir um relatório, apresentar os resultados obtidos, editar um vídeo com o *walkthrough* observado e criar especificações para a implementação da interface.

Esse processo pode ser aplicado em particular nos processos de prototipação em papel. Essa técnica é conhecida pelo seu baixo custo, seu alto potencial de geração de *feedback* e por ser de grande utilidade, além de ser fácil de ser realizada por não-especialistas. Em um trabalho recente, Levi (2005) utilizou a metodologia de prototipagem rápida num Projeto que visa o desenvolvimento de uma aplicação multimídia a ser utilizada por professores do ensino fundamental. A metodologia focou em eliminar problemas relacionados com a apresentação do conteúdo pedagógico e da usabilidade do produto.

A técnica de prototipagem rápida utilizada no ciclo de design iterativo (i.e. concepção, prototipagem e avaliação) da metodologia proposta comprovou-se bastante eficiente. Em apenas duas iterações, a avaliação das primeiras versões do protótipo revelou 56 problemas de usabilidade e 24 novos requisitos que o sistema deverá contemplar.

Segundo o autor, a utilização da prototipagem em papel unificou a visão da equipe sobre as principais características do software em desenvolvimento à medida que apontava lacunas de compreensão a respeito dos conceitos envolvidos. Apesar da grande qualidade do *feedback* gerado a partir das avaliações, a prototipação em papel demandou um custo muito baixo, contabilizadas as horas de trabalho do time-núcleo e os recursos materiais empregados.

#### **4.3.3 Uso de *guidelines* e Guias de recomendação na prototipagem rápida de interfaces educativas**

Há dois tipos de listas de recomendações que podem ser utilizadas em projetos de tais interfaces. O primeiro deles são os guias de estilo ou *guidelines* e o segundo são guias de recomendação. *Guidelines* são mais específicos, pois são editados pelos desenvolvedores de plataformas ou sistemas operacionais, orientando a construção de aspectos visuais e reativos de interfaces gráficas (janelas, botões, barras de rolagem, etc). O objetivo é tornar aplicações consistentes entre si, facilitando a aprendizagem de uso pelos usuários. Recomenda-se que essa observância ocorra em fases iniciais de projeto, em fases de prototipagem. A aplicação de regras de guias de estilo é interessante, pois essas se integram de regras reconhecidas pela comunidade de desenvolvedores como corretas e seu uso já foi observado no design de projetos. Os mais conhecidos são o guia de estilo da IBM, Microsoft (1995), Appel, Linux e Flash.

Os guias de recomendação caracterizam-se pela sistematização de um conjunto de *guidelines* sugestivas para tipos específicos de sistemas interativos independente do tipo de plataforma e visam orientar com boas práticas de projeto a construção das mesmas. Na área de aplicações educativas, Valiati *et al.* (2000) apresentam um guia de recomendações para a construção de aplicações multimídia e um método. Em um estudo recente, os autores discutem dois modelos de avaliação por meio de listas com o objetivo de guiar a construção de software educativo, constituindo-se numa avaliação

processual que tem por objetivo redefinir caminhos para criação de interfaces hipermídia educativa. O método consiste de três etapas: etapa de análise, etapa de elaboração e etapa de avaliação. Na primeira identifica-se o público alvo, delinea-se o tipo de interface e faz-se o planejamento da estrutura preliminar. Na fase de elaboração é feita a triagem de informações a partir de múltiplas fontes para geração das primeiras versões. Na etapa de avaliação, organizam-se estudos de caso nos quais são aplicados os critérios definidos no guia. Os resultados da aplicação do método mostram uma significativa melhora na apresentação das aplicações, garantindo maior controle do usuário sobre a interface, diminuindo a quantidade de erros detectados nas sessões de teste e o aumento expressivo no nível de interatividade e satisfação subjetiva do aluno, durante os ensaios de interação realizados.

#### **4.3.4 Criação de Personas**

A concepção de produtos que pretendem servir milhões de pessoas tem sido um desafio comumente encontrado na indústria de *software*. Para tentar superar as dificuldades de atender às necessidades de inúmeros usuários distintos, a criação e uso de personagens/usuários fictícios e representações concretas, conhecidas como *personas*, apresenta-se como uma interessante técnica de *design* (Grudin e Pruitt 2002). Conforme afirma Nielsen (2002) observar os aspectos sociais e psicológicos dos usuários permite um melhor entendimento das motivações que permeiam as ações desses usuários.

Além de possibilitar a efetiva participação dos usuários no processo de *design*, a técnica de *personas* também reserva uma especial atenção para os aspectos sociais, políticos e culturais característicos do cotidiano dos usuários estudados (Nielsen 2002). Através do engajamento durante um longo período de participantes específicos é possível alcançar a empatia, o comprometimento e uma profunda compreensão do tipo de usuário escolhido. Grudin e Pruitt (2002) apontam ainda para aspectos relativos a valores, medos, aspirações, entre outros atributos que podem ser abordados através da técnica de *persona*. Esta capacidade de investigar tais elementos que em outros métodos passam despercebidos agrega à referida técnica um diferencial importante no processo de *design* interativo.

A técnica de *persona* comumente não é utilizada de forma isolada. Frequentemente aliada ao seu uso utiliza-se a técnica de cenários e outros métodos como cenários (Bødker, 2000). O *persona* também funciona como uma ferramenta eficiente na comunicação de dados que foram coletados através de outros métodos. Segundo Calabria (2004) um aspecto importante da técnica de *persona* é guiar o desenvolvimento do produto a fim de que o conteúdo deste esteja de acordo com os objetivos dos usuários e responda às suas expectativas.

#### **4.4 Avaliação de usabilidade específica para interfaces educativas**

De uma forma geral, testes de usabilidade muito variados podem ser aplicados em versões intermediárias de interfaces educativas e para apresentar tais testes existe uma ampla gama de referências bibliográficas (Hackos e Redish, 1998). Nessa parte descrevemos apenas um referencial teórico e a metodologia de um teste específico utilizado para descrever a relação entre uso da interface (interação) e aprendizagem de conceitos: Análise instrumental (Gomes, 2005).

#### 4.4.1 Análise instrumental

Apresentaremos um modelo teórico e uma metodologia que servem a realização de testes de usabilidade de interfaces de *learnware* para avaliar o potencial de uma interface à aprendizagem de conceitos específicos, matemáticos e científicos, durante sua criação. O modelo foi definido tendo como referencial teórico à abordagem instrumental de Rabardel e a teoria de conceptualização de Vergnaud.

As abordagens de avaliação de software orientadas a diretrizes são úteis para orientar professores na avaliação e seleção de materiais educativos. As recomendações são válidas para uma ampla gama de interfaces, mas não resolvem casos de produtos específicos. Por outro lado, abordagens usadas na indústria, como a análise da tarefa, não permitem analisar a aprendizagem de conceitos específicos.

A proposta da abordagem instrumental é a de apresentar referenciais teórico e metodológico a ser utilizada durante a realização de testes de usabilidade de *learnware*, em ciclos de design iterativo permitindo avaliar se uma interface educativa promove a emergência de conceitos específicos quando usada por usuários representativos.

##### 4.4.1.1 Testes de usabilidade com a abordagem instrumental

Para estabelecer uma relação coerente entre a realização sequencial de tarefas e a aprendizagem de conceitos específicos, um modelo teórico precisa articular dimensões materiais e cognitivas da atividade do usuário com uma interface. Adotou-se nesse trabalho o conceito de instrumento conforme apresentado na teoria da gênese de instrumentos (Mounoud, 1970; Rabardel, 1995).

Segundo estes autores, um instrumento existe quando um esquema mental adapta-se a uma parte específica de uma interface e organiza ações específicas com esse artefato. Artefatos, parte de uma interface, são meios para atingir um objetivo. Os esquemas especializam-se progressivamente, num processo de adaptação contínuo que ocorre por aproximações sucessivas.

O conceito de instrumento permite analisar processos de adaptação do usuário à interface ao longo do tempo de uso da mesma. Em sua definição original, ele ainda não permitiria realizar inferências sobre o conjunto de propriedades de conceitos específicos que são mobilizadas no uso das interfaces; nem sequer encontramos construtos que representem conceitos ou propriedades de conceitos curriculares ou escolares visados na concepção das interfaces.

Uma forma de tornar o conceito de instrumento adequado para responder a esse tipo de pergunta foi substituir da definição original o conceito de esquema mental, colocando em seu lugar a estrutura de esquema mental proposta por Vergnaud (1997). Esse autor descreve a estrutura interna de um esquema mental, inserindo variáveis, que são variadas representações, como as de: objetivos, antecipações das ações, regras de ação, invariantes (teoremas em ato, ou propriedades de conceitos específicos).

Vergnaud propõe que a representação interna a um esquema é composta de informações acerca da realidade que rodeia o usuário e do uso de artefatos (regras de ação), de seus objetivos e conhecimentos identificáveis como científicos, mesmo que implícitos e não explicitáveis pelos usuários (invariantes). As regras são conhecimentos acerca de: (i) aspectos da estrutura de problemas a serem resolvidos, (ii) características, funções e

propriedades do artefato escolhido, (iii) aspectos subjacentes à escolha de um determinado artefato ou, ainda (iv) aspectos subjacentes à interação social e ao contexto material. Os invariantes são conhecimentos que correspondem a conceitos ou propriedades de conceitos matemáticos ou científicos. A hierarquia das categorias usados na definição do modelo está representada na Figura 1.

- Instrumento
  - Esquema instrumental
    - Regras de ação
    - Invariantes
  - Artefato
    - Função do artefato
    - Propriedade do artefato

**Figura 1. Estrutura interna de um instrumento.**

Com essa definição de instrumento temos, então, a possibilidade de finalmente representar e acompanhar a aprendizagem de conceitos específicos que ocorre durante a análise da resolução de um problema, tendo um *learnware* como artefato sobre o qual realiza suas ações.

#### 4.4.1.2 Metodologia: Coleta e análise dos dados

A metodologia aborda um procedimento de avaliação de interfaces no qual as variáveis utilizadas são nominais – teoremas em ato, função dos artefatos, descrição da estrutura dos instrumentos, regras de ação; e o procedimento de análise consiste em analisar a regularidade de emergência de esquemas ou transformações desses elementos, o que permitirá descrever as ações dos usuários com as interfaces e a aprendizagem mediada.

Busca-se encontrar regularidades e transformações na estrutura dos instrumentos constituídos ao longo do tempo. Dados são coletados com procedimentos de observação (captura de tela ou filmagem).

A codificação das ações instrumentais é a primeira etapa do processo de análise. Uma ação regular é associada a um esquema instrumental e a partir daí o especialista infere os elementos que constituem o instrumento a partir de indícios registrados no contexto de resolução do problema com a interface. A partir daí são gerados protocolos de análise qualitativa em forma de tabelas. Nestas, as colunas correspondem as variáveis do modelo de instrumento.

Ações distintas e consecutivas são armazenadas em linhas diferentes e sucessivas da tabela do protocolo. As linhas do protocolo contêm os elementos que constituem ações sucessivas dos usuários. Cada ação é desmembrada em uma série de elementos conforme a definição de instrumento apresentada na seção anterior (ver exemplos em Gomes, 2005). A primeira informação que figura em cada linha da tabela é uma descrição básica da ação esperada, isto é, descreve-se a tarefa que é executada.

Nas colunas seguintes são descritos aspectos do INSTRUMENTO que se constitui no momento da ação do usuário. Descreve-se então sua ORGANIZAÇÃO, em termos da estrutura do esquema que delineia como o usuário usa a interface; e informa-se o ARTEFATO efetivamente utilizado na ação em ocorrência.

Após a realização da coleta de dados, estes podem ser avaliados de várias maneiras. Avalia-se a adaptação do usuário à interface ou a aprendizagem de conceitos específicos. A primeira análise denomina-se análise da gênese instrumental e a segunda denomina-se análise da aprendizagem mediada (Gomes, 1999). Neste artigo não apresentaremos o processo de gênese instrumental, mas apenas a análise da aprendizagem mediada.

#### 4.4.1.3 Análises da adaptação à interface

Em primeiro lugar, analisa-se a seqüência de tarefas dos usuários com a interface, processo similar à técnica tradicional de análise da tarefa. Nesse caso, são analisados a organização do esquema instrumental e os aspectos da estrutura do artefato (função e propriedades do artefato para o usuário). Essa análise permite descrever a forma de adaptação do usuário à interface.

A análise da evolução dos esquemas instrumentais permite sistematizar a organização do esquema mobilizado durante as ações com as interfaces, ao mesmo tempo em que permite registrar os invariantes (propriedades dos conceitos) que são utilizados durante as ações. Assim, é possível comparar diferentes momentos da evolução de um dado esquema. Isso é possível quando representamos sucessivas estruturas de um esquema instrumental ao longo do processo de adaptação à interface.

#### 4.4.1.4 Análise da aprendizagem mediada

A análise da aprendizagem de conceitos específicos é feita mediante a análise dos teoremas-em-ato que são mobilizados em cada uma das tentativas de ação com os artefatos. É possível criar uma representação detalhada da conceitualização que emerge em relação ao uso da interface educativa (Gomes, 1999). Para realizar uma análise da aprendizagem, faz-se necessário inferir acerca dos INVARIANTES (teoremas-em-ato, T.E.A.) subjacentes às ações e as REGRAS de utilização dos artefatos.

A análise da regularidade desses invariantes permite concluir se a interface de um *learnware* influencia na aprendizagem de conceitos específicos.

## 5 Estudo de caso: desenvolvimento de software para ensino de frações

*Nesse estudo de caso descreveremos dois ciclos de iteração do desenvolvimento de um software educativo para o ensino de frações. Os primeiros testes de usabilidade realizados com versões mockups do produto mostraram que a forma de interação da interface não permite a emergência de conceitos relacionados com o conceito de frações, mas de conceitos aditivos. Os resultados levam a modificações significativas na versão já implementada.*

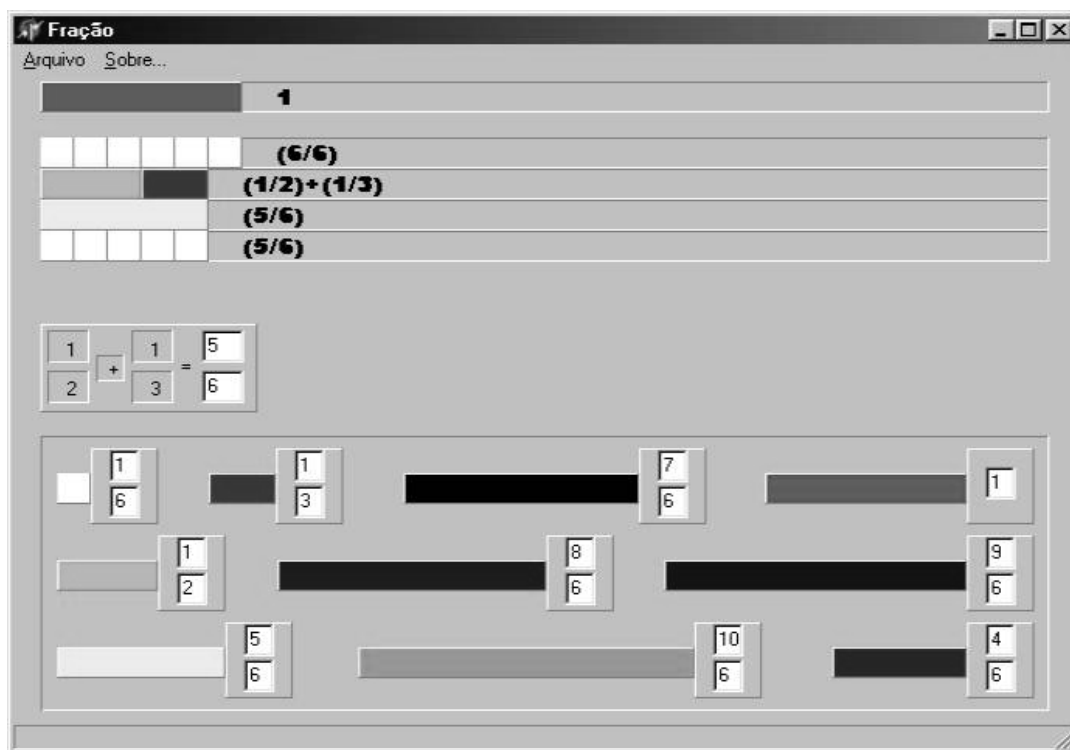
O conceito de fração está associado à idéia de divisão de “um todo” em partes iguais, passível de divisão em partes iguais, até esgotar completamente esse todo considerado (Lima, 1982). Observa-se então que a construção de um significado para o conceito de fração faz necessário que algo seja tomado como unidade - um todo referencial. Esse referencial é, no âmbito social do processo de psicogênese, o resultado de uma negociação entre o professor e os alunos. Há diversas outras propriedades dos conceitos que precisam ser entendidas pelos alunos para que o conceito de fração seja completamente dominado. Fazendo referência a um artigo clássico sobre o assunto na literatura brasileira (Lima, 1982), dizemos com o autor, ser necessário que a criança:

- Entenda que a soma das repartições constituídas no processo de divisão em partes iguais tem que ser percebida pela criança como igual ao todo inicial;
- Perceba a unidade dividida em partes iguais e mesmo assim não alterando a totalidade, nesse caso, “a conservação de quantidade”, é para tanto considerado fundamental. Em suma que a criança perceba a existência de uma totalidade DIVISÍVEL – o todo, constituído pela continuidade (ou agregados, no caso de coleção) e pela descontinuidade das partes;
- Reconheça que uma quantidade permanece invariável através das modificações (repartição, fragmentação, etc.), i.e., ela deve compreender que estas modificações resultam de transformações mentalmente reversíveis o que significa ter a capacidade de entender a ida e volta como aspectos da mesma ação.
- Reconheça a existência de um número determinado de partes (divisão ordenada das partes – cada pessoa corresponde uma parte), relação do todo com suas partes;
- Considere o esgotamento do TODO, i.e., a idéia de que o todo dividido em pedaços até o seu esgotamento sem resultar na existência de RESÍDUOS, que interfere na noção de fração, pois não é uma parte igual à inicialmente retirada, nem é um todo;
- Observe a relação entre o número de partes e o número de cortes – relação das partes com o todo;
- Observe a igualdade das partes (relação entre as partes), a parte se repete como “unidade” em relação ao todo.

A compreensão do conceito de fração como sendo a relação entre uma parte ou partes de um todo em si susceptível de novas divisões implica no fato de constituir mentalmente a representação de uma parte do todo e poder relacionar essa entidade com uma representação do todo. Esse entendimento é possível, mediante o atendimento ao princípio da invariância: a soma das frações constituídas é igual ao todo inicial; concluindo que o conceito de fração resulta do desenvolvimento dessas condições e da coordenação entre as mesmas num conjunto de indícios de antecipação de caráter intuitivo, que orientam a repartição do todo.

### **5.1 Definição da metáfora e Protótipo: Barras de Cuisenaire**

A interface do sistema apresenta a representação de dez quantidades distintas representadas com cores diferentes. A ordem dos tamanhos das peças corresponde às peças dos Blocos de Cuisenaire (Figura 2). Quando utilizamos o material concreto não podemos facilmente realizar a vinculação dos valores numéricos correspondentes a cada quantidade a não ser falando. Isso é bem realizado pelo professor que deve convencer cada aluno dos valores relativos das quantidades e, principalmente, das relações entre as quantidades. O software realiza a função de manter atualizadas as indicações de valores das quantidades relativas. Além disso, essa mesma função entra numa lacuna que há nos materiais disponíveis para o ensino desses conceitos que é a vinculação entre quantidades representadas por áreas e suas representações simbólicas.



**Figura 2 - Tela principal do sistema.**

A tarefa do aluno consiste basicamente em encontrar os valores correspondentes a cada uma das peças e a solução da operação solicitada, quando pertinente. No primeiro caso, o aluno pode deslocar peças diferentes e justapor a peça referente à unidade para, através de estratégias baseadas em estimativas visuais, identificar os valores relativos das peças. Com os valores individuais identificados, o aluno pode então iniciar a resolução da operação. Descrevemos as diferentes operações nas partes seguintes.

## **5.2 Metodologia de teste**

### **5.2.1 Alunos**

Participaram do estudo seis (06) alunos da quinta série de uma escola pública, cada uma com uma dupla de alunos da 5ª série do Colégio de Aplicação de Pernambuco, que tinham entre 11 e 13 anos. Todos os alunos foram voluntários à pesquisa e atenderam a uma solicitação de seu professor de matemática. Os alunos tinham experiência com o computador, pois tinham aulas regulares de informática no colégio. Os alunos que participaram do experimento foram escolhidos aleatoriamente pelo seu professor de matemática e levados para o laboratório de computação do Colégio no qual estudam.

### **5.2.2 Procedimentos**

Como o objetivo era analisar a qualidade do software, não procedemos com qualquer instrução inicial. Apenas explicamos aos alunos como eles poderiam mover as peças, para em seguida solicitar dos alunos a resolução de problemas envolvendo operações com frações. Os alunos foram solicitados a encontrar uma solução para a expressão

proposta na “caixa de expressão” (CE), que são operações envolvendo adição, subtração, multiplicação ou divisão de frações.

### 5.2.3 Material

Os observadores fizeram uso de gravador de áudio, papel e lápis para observações à parte, e um programa para captura de telas (*Screen Cam*) que permite que a tela do computador seja gravada para análise das ações dos participantes da coleta de dados.

## 5.3 Resultados de testes com usuários

A seguir descreveremos as estratégias identificadas em nossa coleta de dados. Foram identificadas diferentes estratégias de uso dos materiais e a partir da análise dos protocolos buscamos associar os conhecimentos sobre frações que possivelmente emergia, conforme nossa interpretação.

### 5.3.1 Estratégia 01: Encontrar o valor das barras que se encontram na “caixa de barras” (CB) mediando com barras

Os alunos, após descobrirem o valor da barra branca, utilizam a mesma como unidade de medida para as demais. Em um exemplo, a dupla 1 encontrou o valor utilizando como estratégia a peça branca. Nessa estratégia a barra branca é usada como unidade de medida. Observa-se que o usuário não necessita pensar em termos de relação parte todo.

[Dupla 01: 530 - 535]

*Agora é a vez da rosa (RS), que sobreposta às 9 brancas (BR), os garotos concluem que RS vale...*

*J- Bote mais pra cá. Dá 10/15 avos.*

*A roxa (RX) é sobreposta às BR, ocupando o espaço de 4 delas.*

*L- Aqui dá 4.*

*J- 4, 4/15 avos.*

Na maioria das vezes preferiram encontrar o resultado da expressão através de cálculo mental e só depois começar a atribuir o valor às barras. Observa-se que os valores atribuídos às peças são valores inteiros e não valores atribuídos a relações entre diferentes partes de peças.

### 5.3.2 Estratégia 02: Encontra o valor da peça branca

Uma das estratégias mais observadas é a determinação do valor da peça branca. A dupla 1 encontravam o valor da peça branca colocando-as uma ao lado da outra na 1ª linha do Quadro de resolução (QR) numa quantidade que correspondesse ao comprimento da peça que equivale ao inteiro na tela.

[Dupla 01: 99 - 103]

*Começam a pôr o valor das barras a partir da BR; Para tanto colocam várias delas, lado a lado, na 1ª linha do Q.C. numa quantidade tal equivalente à unidade dada; Puseram 4.*

*L- 1/4, vai ser 1/4.*

*J- 1/4.*



### 5.3.3 Estratégia 03: Encontra o valor de peças de cor a partir de peça que não a branca

Os alunos procuram identificar os valores das peças do jogo usando como medida o comprimento de uma peça conhecida. Esse método foi mais utilizado pela dupla 02 e esse método foi mais utilizado pela dupla 01.

[Dupla 01: 126 - 128]

Agora a V-E é comparada ainda na C.B. com a PR e depois posta na 3ª linha do Q.C., onde através da visualização, os garotos chegaram à conclusão que seu valor é  $6/4$ ...

[135 - 136]

Apagam todas as barras postas no Q.C. e na sua 1ª linha põem a V-L e comparando-a com a RX, encontram que seu valor é  $3/4$ .

### 5.3.4 Estratégia 04: O valor da branca é igual ao produto dos denominadores

Os alunos concluem, a partir da resolução de alguns problemas, que o valor do denominador da peça branca calculando mentalmente o produto dos valores dos denominadores que aparecem na operação expressa simbolicamente.

## 6 Conclusões

Nesse mini-curso apresentamos uma abordagem centrada no usuário para o desenvolvimento e avaliação de software educativo. Há três aspectos fundamentais nesse tipo de abordagem. Em primeiro lugar, é necessário gerar bom conhecimento sobre o usuário e suas práticas. Num segundo momento do ciclo, esse conhecimento orientará na construção de protótipos variados do produto. Numa terceira etapa do ciclo de projeto são realizados testes de usabilidade. No caso de aplicações educativas, é necessário que os testes permitam captar variáveis relacionadas com a aprendizagem de conceitos. Esses três momentos são repetidos e o refinamento do produto ocorrer por aproximações sucessivas.

## 7 Referencias

- Bastien, J. M., Christian, Scapin, D. L. (1993) *Ergonomic criteria for the evaluation of human-computer interfaces*, Rapport technique no 156. Rocquencourt: INRIA, 1993.
- Bessiere, K., Ceaparu, I., Lazar, J., Robinson, J. e Shneiderman, B. (2002) "Understanding Computer User Frustration: Measuring and Modeling the Disruption from Poor Designs", [www.cs.umd.edu/local-cgi-bin/hcil/sr.pl?number=4409](http://www.cs.umd.edu/local-cgi-bin/hcil/sr.pl?number=4409), Março.
- Bødker S., Scenarios in user-centred design-setting the stage for reflection and action, *Interacting with Computers*, 13 (2000) 61-75.
- Bødker, S. (2000) "Scenarios - setting the stage for reflection and action in user-centered design". *Interacting with computers*, 13 (1): 61-77.
- Boulic, R. and Renault, O. (1991) "3D Hierarchies for Animation", In: *New Trends in Animation and Visualization*, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons Ltd., England.
- Bricker L., *Designs for Computer-Supported Cooperative Learning in Mathematics*. Partial fulfillment of the University of Washington General exam, 1993.

- Brosnan, M.J. (1998) "Technophobia: The Psychological Impact of Information Technology", Routledge, London, In: A.R. Veiga Neto, (1999) "Atitudes de consumidores frente a novas tecnologias". Dissertação de Mestrado, PUC-Campinas.
- Calabria, T. (2004) "An introduction to personas and how to create them", [www.steptwo.com.au/papers/kmc\\_personas/](http://www.steptwo.com.au/papers/kmc_personas/), Março.
- Carraher, D.W. (1990). O que esperamos do Software Educacional? Acesso - Revista de Educação e Informática, Ano II, n. 3, jan./jun. 1990, issn 0103-0736;
- Chapanis, Alphonse. Evaluating usability. In SHACKEL, Brian & RICHARDSON, Simon, eds. Human factors for informatics usability. Cambridge, Cambridge University Press, 1991. p. 359-395
- Christie, B., Scane, R. and Collyer, J., 1995, Evaluation of human-computer interaction at the user interface to advanced IT systems. In Evaluation of Human Work: a Practical Ergonomics Methodology, 2nd edition, edited by J.R. Wilson and E.N. Corlett (London: Taylor & Francis), pp. 310-356.
- Dix, A.; Finlay, J.; Abowd, G.; Beale, R. Human-computer interaction. London: Prentice-Hall Europe, 1998.
- Eason, K. D. Towards the experimental study of usability. *Behaviour and Information Technology*, 3, 1984, 133-143.
- Falcão, T. P. R., (2004) Modelagem de soluções ubíquas para uso em salas de aula no Ensino Fundamental, Trabalho de conclusão do curso de Ciência da Computação – Centro de Informática, UFPE, <Acessado em [www.cin.ufpe.br/~tg](http://www.cin.ufpe.br/~tg)>, <Acessado em 5 de outubro de 2005>.
- Gamez, L. Técnica de inspeção de conformidade ergonômica de software educativo. Dissertação (mestrado em engenharia humana). Minho: Universidade do Minho, 1998.
- Gladcheff, A. P. Um Instrumento de Avaliação da Qualidade para Software Educacional de Matemática. Dissertação (mestrado em Ciência da Computação). São Paulo: Universidade de São Paulo – USP, 2001.
- Gomes, A. S. (1999) Développement conceptuel consécutif a l'activité instrumentée - L'utilisation d'un système informatique de géométrie dynamique au collège, Thèse de doctorat, Université Paris V, Paris, <Disponível em [www.cin.ufpe.br/~tg](http://www.cin.ufpe.br/~tg)>, <Acessado em 5 de outubro de 2005>.
- Gomes, A. S. (2005) *Avaliação da aprendizagem com software educativo no projeto interativo*. Anais do 5º Congresso Internacional de Ergonomia e Usabilidade de Interface Humano Máquina. Rio de Janeiro.
- Gomes, A. S. e Wanderley, E.G. (2003) "Elicitando requisitos em projetos de software educativo". In: WIE 2003, Anais do XXIII Congresso da Sociedade Brasileira de Computação. Campinas: SBC, 2003. v. V. p. 227-238.
- Gomes, A.S., Tedesco, P. e Castro-Filho, J. (2003) "Ambientes de Aprendizagem em Matemática e Ciências". In: Ramos, E., Rosatelli, M. e Wazlawick, R. (Org.). Informática na Escola: um olhar multidisciplinar. Fortaleza, v. 1, p. 108-135.

- Grudin, J. e Pruitt, J. (2002) "Personas, participatory design and product development: an infrastructure for engagement", <Disponível em [www.research.microsoft.com/research/coet/Grudin/Personas/Grudin-Pruitt.doc](http://www.research.microsoft.com/research/coet/Grudin/Personas/Grudin-Pruitt.doc)>, <Acessado em março de 2005>.
- Hackos, J.T. & Redish, J.C. (1998). User and task analysis for interface design. John Wiley & Sons, Inc. Nova Iorque.
- Hinostroza J. E., Mellar H., Pedagogy embedded in educational software design: report of a case study, *Computers & Education* 37 (2001) 27-40;
- Holton, M. and Alexander, S. (1995) "Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials", Computer Graphics: Developments in Virtual Environments, R. A. Earnshaw e J. A. Vince, England, Academic Press, p. 449-460.
- Johnson, D. (1996). "Evaluating the Impact of Technology: the less simple answer", <Disponível em [www.fno.org/jan96/reply.html](http://www.fno.org/jan96/reply.html)>, <Acessado em Maio de 2005>.
- Knuth, D. E. (1984), *The TeXbook*, Addison Wesley, 15th edition.
- Kotonya, Gerald e Sommerville, Ian.: *Requirements Engineering: Processes and Techniques*, John Wiley & Sons, Ltd, 1998.
- Kujala, S., Kauppinen, M., e Rekola, S. (2001). Bridging the Gap between User Needs and User Requirements. In Avouris, N. e Fakotakis, N. (Eds.) Proceedings of the Panhellenic Conference with International Participation in Human-Computer Interaction PC-HCI 2001, Typorama Publications, pp. 45-50.
- Lansdale, M. W.; Ormerod, T. C. *Understanding interfaces: a handbook of human-computer dialogue*. London: Academic Press, 1995.
- Leont'ev, A.N. (1975). Actividad, conciencia, personalidad. Habana: Editorial Pueblo y Educación, edition in Spanish 1981, edited from original in Russian 1975.
- Levi, F. L. B., (2005) Prototipação rápida no ciclo de design iterativo de aplicações multimídia para formação de professores, Trabalho de conclusão do curso de Ciência da Computação – Centro de Informática, UFPE, <Acessado em [www.cin.ufpe.br/~tg](http://www.cin.ufpe.br/~tg)>, <Acessado em 5 de outubro de 2005>.
- Lima, M. F. (1982). A conservação de quantidades e iniciação ao conceito de frações. Em Carraher, T. N. Org., *Aprender Pensando*. Editora Vozes. Petrópolis – RJ.
- Lins, W. C. (2004) "Análise da atividade docente com software educativo no contexto do laboratório de informática". Dissertação de Mestrado, UFPE.
- Lins, W. C. e Gomes, A. S. (2003) "Educational software interfaces and teacher's use". HCIi 2003, Creta, Grécia, New Jersey: Lawrence Erlbaum, v. 1, p. 971-975.
- Lyra, A. R. L.; Leitão, D. A.; Amorim, G. B. C.; Gomes, A. S. Ambiente virtual para análise de software educativo. WIE 2003. Campinas, SP: SBC, 2003.
- Maguire M., Methods to support human-centred design, *Int. J. Human-Computer Studies* (2001) 55, 587-634.
- Marcus, Aaron. Graphical User Interfaces – rules of thumb. In HELANDER et alli, eds. *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier, 1997. p. 438-438.

- Martins L. E. G. e Daltrini B. M., Utilização dos preceitos da Teoria da Atividade na Elicitação dos Requisitos do Software, SBES'1999.
- Mayhew, D., *Principles and guidelines in software user interface design*. New Jersey: Prentice Hall, 1992.
- McDougall, A. e Squires, D. (1995) An Empirical Study of a New Paradigm for Choosing Educational Software, *Computers and Education*, (25) 3, pp. 93-103.
- Moraes, A. de. Ergonomia: Usabilidade de Interfaces, Interação Humano-Computador, Arquitetura da Informação. In Anais do 2o USIHC. Rio de Janeiro: PUC-Rio, 2003.
- Nardi, B.A.(1996) *Activity Theory and Human-Comp. Interaction*. London: MIT Press.
- Nielsen, J. *Usability engineering*. San Diego: Morgan Kaufmann, 1995.
- Nielsen, L. (2002) "From user to character – an investigation into user-descriptions in scenarios", <http://web.cbs.dk/staff/lene/from%20user%20to%20character.pdf>, Junho.
- Preece, J.; Rogers, Y.; Sharp, H. *Interaction design*. NY: John Wiley & Sons, 2002.
- Rabardel P., *Les hommes et les technologies - Approche cognitive des instruments contemporains*, Paris: Armand Colins, 1995.
- Ravden, S.; Johnson, G. *Evaluating usability of human-computer interfaces*. Chichester: Ellis Horwood, 1989.
- Shackel, Brian. Usability – context, framework, definition, design and evaluation. In. SHACKEL, Brian & RICHARDSON, Simon, eds. *Human factors for informatics usability*. Cambridge, Cambridge University Press, 1991. p. 21-38.
- Shneiderman, Ben. *Designing the user interface: Strategies for effective human-computer interaction*. Maryland: Addison-Wesley, 1998.
- Tchounikine P., Pour une ingénierie des Environnements Informatiques pour l'apprentissage humain, *Information-Interaction-Intelligence*, vol. 2, n. 1, 2002.
- Valente, J. A. Análise dos diferentes tipos de softwares usados na educação. In Anais do III Encontro Nacional do PROINFO. Pirenópolis: MEC, 1998.
- Veiga Neto, A.R. (1999) "Atitudes de consumidores frente a novas tecnologias". Dissertação de Mestrado, Campinas, SP, Brasil, PUC-Campinas.
- Vergnaud G. (1997) The nature of mathematical concepts. In T. Nunes et P. Bryant (Eds.) *Learning and teaching mathematics: An international Perspective* (1997), pp. 5-28, Psychology Press, Hove;
- Windows Styleguide (1995) *The Windows Interface Guideline – A Guide for Designing Software*, Microsoft Corporate.
- Balacheff, N., Kaput, J., Computer based Learning Environments in Mathematics. Editor(s): Bishop, Alan J.; Clements, Ken; Keitel, Christine; Kilpatrick, Jeremy; Laborde, Colette, Dordrecht: Kluwer. 1996, p. 469-501;
- Schwartz J. L., Can technology help us make the mathematics curriculum intellectually stimulating and socially responsible? *International Journal of Computers for Mathematical Learning* 4: 99–119, 1999.